CLAIMS

What is claimed is:

1. A method for demultiplexing an incoming packet to a virtual machine ("VM"), comprising:

   unmapping a guest physical address from a host physical address in at least one page table entry associated with buffers in a direct memory access ("DMA") table to create unmapped buffers;

   placing the incoming packet into at least one of the unmapped buffers; and

   allocating the at least one of the unmapped buffers to the VM to create a mapped buffer.

2. The method according to Claim 1 wherein unmapping the guest physical address from the host physical address further comprises clearing the contents of a physical page associated with the host physical address.

3. The method according to Claim 1 wherein allocating the at least one of the unmapped buffers further comprises temporarily assigning the at least one of the unmapped buffers to the VM to create the mapped buffer.

4. The method according to Claim 1 further comprising:

   causing the VM to release the mapped buffer; and

   unmapping the guest physical address from the host physical address.

5. The method according to Claim 4 wherein causing the VM to release the mapped buffer further comprises injecting a signal into the VM.

6. The method according to Claim 5 wherein the signal is an interrupt.

7. A method for demultiplexing an incoming packet to multiple VMs, comprising:

   decoupling a guest physical address for a virtual machine ("VM") from a host physical address to create unmapped buffers;

placing incoming packets in the unmapped buffers;

examining the incoming packets to determine appropriate destination VMs; and

assigning the unmapped buffers to the appropriate destination VMs.

8. The method according to Claim 7 wherein decoupling the guest physical address from the host physical address further comprises invalidating entries in at least one page table entry for buffers in a direct memory access table associated with the VM.

9. A system for demultiplexing an incoming packet to an appropriate virtual machine ("VM"), comprising;

a plurality of VMs;

a component coupled to the plurality of VMs, the component capable of invalidating entries in at least one page table entry for direct memory access ("DMA") buffers to create unmapped buffers, placing the incoming packet in the unmapped buffers, determining which of the plurality of VMs is the appropriate destination virtual machine ("VM") for the incoming packet and assigning the unmapped buffers with the incoming packet to the appropriate destination virtual machine.

10. The system according to Claim 9 wherein the component is one of a demultiplexer and a virtual network interface card ("VNIC").

11. The system according to Claim 10 wherein the VNIC is maintained by a virtual machine manager ("VMM") coupled to the plurality of VMs.

12. An article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to demultiplex an incoming packet to a virtual machine ("VM") by:

unmapping a guest physical address from a host physical address in at least one page table entry for buffers in a direct memory access ("DMA") table to create unmapped buffers;

9

placing the incoming packet into at least one of the unmapped buffers; and

allocating the at least one of the unmapped buffers to the VM to create a mapped buffer.

13. The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to unmap the guest physical address from the host physical address further by clearing the contents of a physical page associated with the host physical address.

14. The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to allocate the at least one of the unmapped buffers by temporarily assigning the at least one of the unmapped buffers to the VM to create the mapped buffer.

15. The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to demultiplex an incoming packet by: causing the VM to release the mapped buffer; and unmapping the guest physical address from the host physical address.

16. The article according to Claim 15 wherein the instructions, when executed by the machine, further cause the VM to release the mapped buffer by injecting a signal into the VM.

17. The article according to Claim 16 wherein the instructions, when executed by the machine, further cause the VM to release the mapped buffer by injecting a signal into the VM.

18. The article according to Claim 17 wherein the instructions, when executed by the machine, further cause the VM to release the mapped buffer by injecting an interrupt into the VM.

19. An article comprising a machine-accessible medium having stored thereon
    instructions that, when executed by a machine, cause the machine to
    demultiplex an incoming packet to multiple VMs by:
    decoupling a guest physical address for a virtual machine ("VM") from a host
        physical address to create unmapped buffers;
    placing incoming packets in the unmapped buffers;
    examining the incoming packets to determine appropriate destination VMs; and
    assigning the unmapped buffers to the appropriate destination VMs.

20. The article according to Claim 19 wherein the instructions, when executed by
    the machine further decouple the guest physical address from the host physical
    address further by invalidating entries in a direct memory access table
    associated with the VM.